Af IDW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:

Kenneth A. Ward

Serial No. 09/553,966

Filed: April 21, 2000

For: MECHANISM FOR EFFICIENT SCHEDULING OF COMMUNICATION FLOWS

§ Group Art Unit: 2665
§
§ Examiner: Philpott, Justin M.
§
§ Atty. Dkt. No.: 5181-36000
§ P4233

CERTIFICATE OF MAILING
37 C.F.R. § 1.8

I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below:

Robert C. Kowert
Name of Registered Representative

January 10, 2005
Date

Signature

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Appeal filed November 12, 2004, Appellants present this Appeal Brief. Appellants respectfully request that the Board of Patent Appeals and Interferences consider this appeal.

01/14/2005 SSESHE1 00000062 501505 09553966
01 FC:1402 500.00 DA

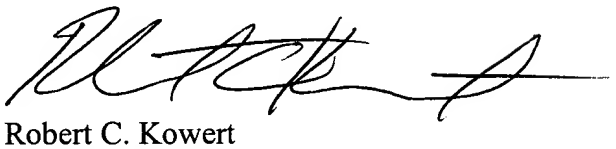
Christensen indicates whether there are any interrupt request entries needing services (e.g. with the PDN register).

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 12-38 was erroneous, and reversal of his decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of $500.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-36000/RCK. This Appeal Brief is submitted with a return receipt postcard.

Respectfully submitted,



Robert C. Kowert
Reg. No. 39,255
Attorney for Appellants

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8850

Date: January 10, 2005

I. REAL PARTY IN INTEREST

As evidenced by the assignment recorded at Reel 010745, Frame 0389, the subject application is owned by Sun Microsystems, Inc., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and now having its principal place of business at 4150 Network Circle, Santa Clara, CA 95054.

II. RELATED APPEALS AND INTERFERENCES

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-11 are allowed. Claims 12-38 stand finally rejected. The rejection of claims 12-38 is being appealed. A copy of claims 12-38 is included in the Claims Appendix herein below.

IV. STATUS OF AMENDMENTS

An amendment to claim 18 was filed on October 12, 2004, but was not entered by the Examiner, as noted in the Advisory Action of November 26, 2004. All references to claim 18 herein refer to the state of claim 18 prior to the October 12, 2004 submission. No other amendments to the claims have been submitted subsequent to the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

A common theme in modern I/O architectures is the need to service many communication requests to many different communication channels that are sharing a limited total bandwidth. One problem is how to divide or allocate the available bandwidth amongst the various communication requests. Typically the number of

channels greatly exceeds the number of physical media paths that are shared by the physical devices to which communication requests are being made. In conventional systems, just determining which channel has a pending request that should be serviced next may be time consuming.

One prior art solution to allocating bandwidth amongst the channels is a round robin, one message per channel allocation. However, a message-based allocation may unfairly favor requestors with larger messages. Therefore, it may be desirable to have a more fair bandwidth allocation than a pure one message per channel mechanism. It may also be advantageous to group classes of channels together in order to vie for gross amounts or percentages of media bandwidth. This combination of channel grouping into classes and bandwidth allocation based on class may provide for differing levels of service.

Independent claim 12 is directed to a method for servicing a plurality of communication channels. As discussed on page 7, lines 15-17 and page 12, lines 5-15, such a method may include, for a first service time, selecting a set bit in a top level of a hierarchical channel map. The set bit in the top level indicates a group of bits in the next level of the hierarchical channel map to examine. *See also* FIGs. 3 and 4. The hierarchical channel map may include one or more intermediate levels. The top level of the hierarchical channel map may be a previous level for a first one of the one or more intermediate levels of the hierarchical channel map. The method may also include examining in each of one or more intermediate levels of the hierarchical channel map only a group of bits indicated by the set bit selected in a previous level and selecting a set bit from each examined group. *See, e.g.*, FIGs. 3, 4 and 5; page 7, lines 17- 20; page 11, line 23 – page 12, line 3; page 12, lines 14-15.

The method may further include examining in a lowest level of the hierarchical channel map only a group of bits indicated by the previous intermediate level and selecting a set bit from the examined group of the lowest level. The selected bit at the

lowest level indicates one of the plurality of communication channels to be serviced. *See, e.g.*, FIGs. 3, 4 and 5; page 6, lines 9-11; page 7, lines 4-7, and lines 20-24; page 11, lines 7-21. The method may also include servicing a communication request from the communication channel indicated by the selected bit from the lowest level of the hierarchical channel map, for example, as described in the specification at page 7, lines 11-13 and lines 22-24. *See also* FIGs. 3, 4, 5, page 8, lines 4-6; page 12, line 26 – page 13, line 4.

Independent claim 18 is directed to a system for servicing communication queues. The system may include memory configured to store a hierarchical channel map that has a plurality of levels. Each bit of the lowest level is mapped only to a different one of a plurality of communication channels, and each bit of each higher level is mapped to a group of bits at the next lower level. *See, e.g.*, FIGs. 3 and 4; page 6, lines 4-15; page 7, lines 15-17; page 11, lines 3-21; page 12, lines 5-15. Such a system may also include a host adapter configured to maintain the hierarchical channel map. Each bit at the lowest level is set if the channel to which it is mapped has a pending communication request and is cleared if not, and each bit of each higher level is set if at least one bit is set in the lower level group to which it is mapped and is cleared if not. *See, e.g.*, page 7, line 30 – page 8, line 4; FIG 3, 4, 8. The host adapter may also be configured to examine the hierarchical channel map to determine a next one of the communication channels to service. *See, e.g.*, page 8, lines 4-6; page 12, line 26 – page 13, line 4.

Independent claim 29 is directed to a computer readable medium having program instructions operable to select a first bit set in a current top level group of a hierarchical map after a position indicated by a top level service mask. *See, e.g.*, FIGs. 4 and 5 – item 302; page 14, lines 17-26; page 14, line 28-page 15, line 1. The program instructions are further operable to set the top level service mask to indicate the position of the selected first bit in the current top level group and in a second level of the hierarchical channel map, access a second level group indicated by the selected set bit from the top level. *See,*

e.g., FIG. 5, item 304; page 15, lines 1-3.

The program instructions are further operable to select a first bit set in the accessed second level group after a position indicated by a second level service mask; set the second level service mask to indicate the position of the selected bit in the accessed second level group; and in a bottom level of the hierarchical channel map, access a bottom level group indicated by the selected set bit from the second level. *See, e.g.*, FIG. 5, items 306, 308, 310 and 312; page 15, lines 2-6. The program instructions are further operable to select a first bit set in the accessed bottom level group after a position indicated by a bottom level service mask; set the bottom level service mask to indicate the position of the selected bit in the accessed bottom level group; and service a request from a channel indicated by the selected set bit from the bottom level. *See, e.g.*, FIG. 5, items 314, 316 and 318; page 15, lines 5-9; FIG. 7, page 16, lines 2-12.

Independent claim 30 is directed to a method for servicing a plurality of communication channels, similar to the method recited by claim 12, discussed above. The method recited by claim 30 includes that the groups at each intermediate level are sized smaller than at the previous intermediate level as illustrated by FIGs. 3 and 4. The method recited by claim 30 further includes examining a top level of a hierarchical channel map to select a section of the communication channels in which at least one channel has a pending communication request. The top level indicates for each of a plurality of sections of the communication channels if at least one channel of that section has a pending communication request. *See, e.g.*, FIGs. 3 and 4; page 7, lines 15-17 and page 12, lines 5-15.

The method recited by claim 30 also includes, for the first service time, examining a portion of one or more intermediate levels of the hierarchical channel map to select a lowest level group of the communication channels in which at least one channel has a pending communication request. Each intermediate level indicates for each of a plurality of groups of the communication channels if at least one channel of that group has a

pending communication request. The groups at each intermediate level are sized smaller than at the previous intermediate level, and examining a portion of each intermediate level determines which portion of the next hierarchical channel map level to examine. *See e.g.*, FIGs. 3, 4 and 5; page 7, lines 17- 20; page 11, line 23 – page 12, line 3; page 12, lines 14-15. The method recited by claim 30 also includes selecting a next communication channel to be serviced from the lowest level group selected by examining a portion of one or more intermediate levels. *See, e.g.*, FIGs. 3, 4 and 5; page 6, lines 9-11; page 7, lines 4-7, and lines 20-24; page 11, lines 7-21.

Independent claim 31 is directed to a system for servicing a plurality of communication channels including a first memory configured to store one or more levels of a hierarchical channel map. The one or more levels include a lowest level for storing an indication of which ones of the plurality of communication channels have pending communication requests. *See, e.g.*, FIGs. 4, 5 and 6; page 14, lines 17-26; page 14, line 28-page 15, line 1.

The system may also include a second memory configured to store a top level of the hierarchical channel map. The plurality of communication channels are organized in channel sections, and for each channel section the top level indicates if at least one of the communication channels within that section has a pending communication request. *See, e.g.*, FIG. 9; page 17, lines 6-12; page 18, lines 14-30.

The system may also include a host adapter configured to determine a next channel to service by examining the lowest level in the first memory. The host adapter determines the next channel to service by examining no more of the lowest level than a portion of the lowest level corresponding to one of the channel sections indicated by the top level as having at least one pending communication request. *See, e.g.*, FIGs. 3, 4 and 8; page 7, lines 15-24, and line 30 – page 8, line 6; page 12, line 26 – page 13, line 4.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 18-23 and 25-28 stand finally rejected under 35 U.S.C. § 102(b) as being anticipated by Christensen et al. (U.S. Patent 4,271,468) (hereinafter "Christensen").

2. Claims 12-17, 24 and 29-38 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Christensen.

VII. ARGUMENT

First Ground of Rejection:

Claims 18-23 and 25-28 are rejected under 35 U.S.C. § 102(b) as being anticipated by Christensen et al. (U.S. Patent 4,271,468) (hereinafter "Christensen"). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 18, 22 and 24:

Regarding claim 18, Christensen does not anticipate a memory configured to store a hierarchical channel map comprising a plurality of levels, wherein each bit of the lowest level is mapped only to a different one of a plurality of communication channels, and wherein each bit of each higher level is mapped to a group of bits at the next lower level.

Christensen teaches a system for handling channel interrupts where interrupt requests (IR) are put in to queues Q0-Q7 and a plurality of central processors (CP) coordinate with a system controller (SC) to remove and service the IR entries from the queues (Abstract, column 3, lines 5-25). Christensen also teaches the use of a single pending register (PND) in which each bit of the PND register indicates whether a respective queue Q0-Q7 is empty or contains a pending (non-serviced) interrupt request.

The Examiner equates Christensen's queues Q0-Q7 stored in main storage 23 to the lowest level of the hierarchical channel map recited in appellants claim 18. However, claim 18 recites that *each bit* of the lowest level is mapped *only to a different one* of a plurality of communication channels. In contrast, Christensen's queues Q0-Q7 only function as queues to store interrupt requests. There is no teaching in Christensen that each bit in each of queues Q0-Q7 is mapped *only to a different one* of a plurality of communication channels. As taught in Christensen, each queue entry stores a channel interrupt request until the request is removed from the queue by one of the central processors (col. 7, lines 20-27). In Christensen, once an interrupt request has been removed from a queue, the same I/O channel may or may not have another interrupt request pending in a different entry of the same queue or even a different queue. Furthermore, in Christensen, once an interrupt request has been removed from a queue, the same queue entry can be used to store an interrupt request from a completely different I/O channel. Thus, each of Christensen's queues Q0-Q7 is not mapped *only to a different one* of a plurality of communication channels, let alone *each bit* being mapped only to a different one of a plurality of communication channels.

Additionally, each of Christensen's queues may easily contain multiple entries for the same communication channel (column 6, lines 1-17). In fact, Christensen teaches that an interrupt request is placed on a particular queue according a prearrangement of queues to the source and type of interrupt (column 3, lines 11-15). In other words, similar interrupts from different sources and interrupts from the same source or channel are placed onto the same queue. Thus, there is no possible interpretation of Christensen that allows queues Q0-Q7 to be a lowest level of a hierarchical channel map wherein <u>each bit is mapped only to a different one of a plurality of communication channels</u>.

Thus, Christensen's queues Q0-Q7 clearly do not store a hierarchical channel map comprising a plurality of levels, wherein each bit of the lowest level is mapped only to a different one of a plurality of communication channels, and wherein each bit of each higher level is mapped to a group of bits at the next lower level, as recited in claim 18.

In further regard to claim 18, Christensen also fails to teach a host adapter configured to maintain the hierarchical channel map, wherein each bit at the lowest level is set if the channel to which it is mapped has a pending communication request and is cleared if not. Instead, Christensen teaches that interrupt request entries are stored and removed from queues Q0-Q7. Christensen does not mention that each bit of queues Q0-Q7 is set if the channel to which it is mapped has a pending communication request and is cleared if not. There is no such setting and clearing of bits in Christensen's queues. Furthermore Christensen does not map individual bits to channels. Under Christensen's system, there is no single specific bit that reflects the state (e.g. whether there is a pending interrupt request) of any *particular* channel. Instead, Christensen's queues function purely and only as queues storing interrupt request entries. Entries are added as interrupts occur and removed as they are serviced.

Furthermore, Christensen does not teach a host adapter that is configured to examine the hierarchical channel map to determine a next one of the communication channels to service, as recited in claim 18. The Examiner equates Christensen's channel controller SC24 with the host adapter of claim 18. However, SC24 does not examine a hierarchical channel map to determine a next channel to be serviced. Christensen teaches that interrupt requests are added to the queues by SC24 and removed by the central processors. Adding interrupt request entries to a queue is not an examination of a hierarchical channel map to determine a next channel to be serviced.

The Examiner contends that PND register 46 is the top level of a hierarchical channel map. However, Christensen's channel controller 24 does not examine pending (PND) register 46 to determine a next channel to be serviced. Also, contrary to the Examiner's assertion, the order in which interrupt requests are placed on the queues does not determine the order in which they are serviced. Instead, Christensen teaches that the state of PND register 46 is continuously broadcast to the central processors (col. 6, lines 60-62). The central processors select a queue corresponding to one of the states broadcast

by the system controller 22. A central processor indicates an accepted queue to the system controller 22 (col. 6, line 62 - col. 7, line 5). Only entries in a single queue are serviced in the order that they are placed on that queue. For a particular queue, the system controller 22 then selects one central processor to service the first entry in that queue from among all processors that accepted that queue (col. 7, lines 30-59). Thus, no component in Christensen's system ever examines either PND register 46 or queues Q0-Q7 to determine a next channel to be serviced. Instead, Christensen's system controller 22 selects a central processor to handle a single entry from each queue by the method described above.

Claim 19:

Christensen fails to anticipate a system wherein the host adapter is configured to service for one service unit a channel request from a channel mapped to a set bit at the lowest level of the hierarchical channel map, wherein the set bit is selected by examining a current group of bits at the lowest level of the hierarchical channel map to select a next set bit in the group indicating a channel with a pending request, and if no more bits are set in the current group, examining a current group at the next higher level to select a next set bit and then examining the next lower level group indicated by the selected higher level set bit.

The Examiner argues, without cited any particular portion of Christensen, that Christensen's system includes selecting a next entry within a designated queue among queues Q0-Q7. However, the Examiner's interpretation of Christensen's system is incorrect. As described above regarding claim 18, Christensen does not teach a hierarchical channel map. Instead, Christensen teaches that system controller SC22 broadcasts the state of each of the bits in the interrupt queue pending (PND) register in parallel to each CP. Each CP that is in an interruptible state responds and accepts one of the queues Q0-Q7 to potentially service (column 3, line 45 – column 4, line2). CC24 then selects a CP to service the front entry in a queue. Christensen does not teach servicing a channel by selecting a set bit to which the respective channel is mapped. Nor

does Christensen teach selecting the set bit by examining a current group of bits *at the lowest level* of the hierarchical channel map.

The Examiner argues that Christensen's queues Q0-Q7 are the lowest level of a hierarchical channel map. However, following the Examiner's interpretation, Christensen's system still does not examine a group of bits in one of queues Q0-Q7 to select *a next set bit* indicating a channel with a pending request. This is clearly not taught by Christensen nor performed by his system. Instead, Christensen clearly teaches that the entries of each of queues Q0-Q7 "are processed in a first-in, first-out basis" (column 3, lines 22-25). Thus, there is no need in Christensen's system to select a next set bit in a current group of the lowest level of the hierarchical channel map since the entries of Christensen's queues are always processed in order. The first-in, first-out processing of queue entries is well understood in the art and does not involve examining a current group of bits at the lowest level of a hierarchical channel map to select a next set bit, as recited in claim 19. Not only has the Examiner failed to point out any portion of Christensen describing the examination of groups of bits in queues Q0-Q7, there is no need for such an examination in Christensen's system. The Examiner is merely using hindsight analysis to insert the limitations of claim 19 into Christensen's system – which is clearly improper.

Additionally, Christensen describes how after a CP completes the processing of one entry from a queue, the CP cleans up and returns to normal processing. Any additionally entries in the same queue are handling in the same manner as the first entry – e.g. by CS22 broadcasting the state of the PND register and a CP (not necessarily the same CP that serviced the previous entry) responding by accepting that queue. Thus, nothing in Christensen's system ever examines the contents of any of queues Q0-Q7 to select a next set bit in the group indicating a channel with a pending request.

Claim 20:

Christensen fails to teach a service mask for each level of the hierarchical channel map, wherein each service mask is configured to indicate the next bit position in the current group to be examined for a set bit. The Examiner argues that Christensen's I/O mask 51 is a service mask. However, I/O mask 51 does not indicate a next bit position in the current group to be examined for a set bit. Instead, Christensen teaches, "[a]ny broadcast bit PND(O)-(PND(7) may be masked off by a CP with the conventional I/O mask bits in a control register 51 in the CP" and that "[a]ny PND bit which is masked off is not available for acceptance" (column 6, lines 56-59). Thus, I/O mask 51 prevents individual queues from being available for servicing by specific central processors. This is quite different from indicating the next bit position in a current group of bits to be examined. No bit position is ever indicated by I/O mask 51.

Furthermore, Claim 20 recites a service mask for each level of the hierarchical channel map. The Examiner maintains (erroneously) that Christensen's queues and PND register make up a hierarchical channel map and that I/O mask 51 is a service mask. However, Christensen teaches a separate I/O mask 51 in the interrupt acceptance circuits for each of multiple central processors (FIG. 3A, item 51 and column 8, lines 58-65). Thus, the Examiner's interpretation of Christensen's I/O mask 51 as a service mask for each level of a hierarchical channel map cannot be correct. Rather than having an I/O mask 51 for the queues Q0-Q7 and another I/O mask 51 for the PDN register, as would follow from the Examiner's interpretation, Christensen teaches having a separate I/O mask 51 for each central processor.

Claim 21:

Christensen fails to anticipate a system wherein each service mask is configured to indicate the bit position within the current group for the corresponding level of the last selected set bit in that group, as recited in claim 21. The Examiner maintains, as described above regarding claim 20, that Christensen's I/O mask 51 is a service mask and further contends that I/O mask 51 indicates a bit position in the PND register, citing column 6, lines 51-68 of Christensen. However, I/O mask 51 does not indicate any bit

position of the PND register, and certainly does not indicate the bit position of the last selected set bit in the current group of a corresponding level of the hierarchical channel map, as recited in claim 21. Instead, I/O mask 51 is used to prevent a particular central processor from accepting, and thus servicing, specific queues as is discussed at the Examiner's cited passage (*See also* FIG. 3A, item 51 and column 8, lines 58-65, and column 6, lines 56-59 and lines 62-64). Furthermore, I/O mask 51 does not indicate anything regarding the last set bit (or the last serviced entry or queue).

Christensen further fails to anticipate a system wherein the host adapter is configured to examine each group for the next set bit after the bit position indicated by the corresponding service mask, as recited in claim 21. The Examiner argues that Christensen's channel controller CC24 is a host adapter as recited in claim 21 and cites column 6, lines 24-50. However, the cited passage only describes how CC24 puts interrupt request entries in queues Q0-Q7 and updates the PDN register to reflect the state (empty or non-empty) of each of the queues. Christensen does not teach that CC24 examines any groups of bits in queues Q0-Q7 for a next set bit after the bit position indicated by the corresponding service mask, which the Examiner argues is I/O mask 51. CC24 does not determine a next set bit after a bit position indicated by I/O mask 51. In fact, Christensen does not teach that CC24 ever examines I/O mask 51 for any reason whatsoever.

Thus, Christensen clearly fails to anticipate a system wherein each service mask is configured to indicate the bit position within the current group for the corresponding level of the last selected set bit in that group, wherein the host adapter is configured to examine each group for the next set bit after the bit position indicated by the corresponding service mask, as recited in claim 21.

Claim 23:

Christensen fails to anticipate a system wherein each group of bits at the lowest level is accessible by a single memory access, as recited in claim 23. The Examiner

contends that Christensen's queues Q0-Q7 are accessible by a single memory access, but cites column 9, line 24 – column 10, line 21 where Christensen describes the interrupt acceptance determination circuits of his central processors (CPs). However, the cited portion fails to mention anything regarding the reading of queues Q0-Q7 or how the groups of bits in queues Q0-Q7 are accessible by a single memory access.

The Examiner is clearly speculating regarding how queues Q0-Q7 are accessed in Christensen's system.

Claim 25:

Christensen fails to anticipate wherein the host adapter is configured to service channel requests according to service classes, wherein each channel is mapped to one or more service classes, and wherein the next channel to be serviced is selected by examining only portions of said hierarchical channel map corresponding to channels mapped to a current service class.

Firstly, the Examiner has not provided a proper rejection of claim 25. The Examiner argues that Christensen teaches a plurality of service class masks, wherein each service class mask is configured to map a number of the communication channels to one of a plurality of service classes, wherein each service class is allocated a portion of the limited bandwidth on the communication fabric. However, claim 25 does not recite the limitations referenced by the Examiner. Thus, the Examiner's rejection of claim 25 is clearly improper.

Furthermore, Christensen's I/O mask 51 is not a service class mask as suggested by the Examiner. Instead, I/O mask 51 controls which of the central processors CPs can service which each of queues Q0-Q7. Additionally, Christensen teaches a separate I/O mask 51 in the interrupt acceptance circuit for each central processor not for service classes. Thus, I/O mask 51 does not represent a mapping of channels to service classes,

but instead, controls which queues a particular CP is allowed to service. Please also refer to the discussion of I/O mask 51 regarding the rejection of claim 20, above.

Christensen does not describe selecting the next communication channel to service by examining only portions of a hierarchical channel map corresponding to channels mapped to a current service class. I/O mask 51 (which the Examiner contends is a service class mask) does not have any bearing on which queue Q0-Q7, or which entry of a queue, should be serviced next, but instead only controls the queues that may be serviced by any particular CP. I/O mask 51 may prevent a particular CP from servicing a queue, but it has no bearing on the order in which the queues Q0-Q7 are serviced.

Claim 26:

Christensen fails to anticipate wherein the current service class is selected according to an order of service classes indicated by a service array.

Firstly, similar to the rejection of claim 25 above, the rejection of claim 26 is improper because the Examiner has presented arguments regarding features not present in claim 26. Specifically, in the rejection of claim 26, the Examiner refers to the limitations of claim 10, which are different than the limitation recited in claim 26. Thus the rejection of 26 is improper.

Furthermore, the Examiner argues that Christensen's QID register 82 is a service array, citing column 17, lines 22-30 and column 9, line 24- column 10, line 31. However, Christensen clearly describes how QID register 82 "is set to the value of the lowest-numbered enabled PDN output line having a one state" (column 9, lines 27-30). Christensen also teaches that the lowest numbered PDN output line corresponds to the highest priority queue (column 9, lines 16-24). Thus, QID register 82 temporarily stores a queue identifier (hence QID) used as part of SC22's determination of which CP should service the queue identified by QID register 82. *See also* column 6, line 60 – column 7, line 29; and column 9, lines 24-30. Thus, QID register 82 is a register that indicates a

queue, not a service array indicating an order of service classes according to which a current service class is selected.

Additionally, the Examiner maintains that the priority of Christensen's queues are services classes and that QID register 82 is a service array indicating an order of service classes. However, QID register 82 is set to a value representing the highest priority queue. Thus, rather than QID register 82 (interpreted as a service array) indicating the order of priority (interpreted as service classes) for the servicing queues Q0-Q7, QID register 82 only indicates the single current queue for which SC22 selects a CP to service. Since QID register 82 only stored a single queue identifier, QID register 82 cannot possible be considered an *array* that indicates an *order of service classes* (plural).

Claim 27:

Christensen fails to anticipate one service mask per level of the hierarchical channel map per service class, wherein each service mask is configured to indicate the next bit position in the current group to be examined for a set bit when the corresponding service class is the current service class.

As with the rejections of claims 25 and 26, discussed above, the Examiner has failed to present arguments regarding the specific limitations of claim 27, but instead has presented arguments regarding the limitations of claim 11, which differ from those of claim 27. Thus, the rejection of claim 27 is improper.

Additionally, the Examiner argues that I/O mask 51 is a service mask indicating a next bit position to be examined. As noted above, Christensen's I/O mask 51 does not indicate a next bit position to be examined within a selected group of bits to determine a selected group of bits to be examined at the next level, as suggested by the Examiner (Please refer to the discussion regarding the rejection of claim 20). Instead, as described above, Christensen teaches a separate I/O mask 51 in the interrupt acceptance circuit for each CP that only controls which of queues Q0-Q7 a particular CP is allowed service.

Thus, the Examiner's line of reasoning fails to provide a service mask per level of a hierarchical channel map (plus, as argued above, Christensen does even teach a hierarchical channel map).

The Examiner also argues that I/O mask 51 indicates the next bit position to be examined within the designated non-empty queue having the highest priority. However, as discussed above, I/O mask 51 does not indicate anything within a particular queue, but instead merely allows the masking of entire queues from being accepted by, and therefore from being serviced by, individual central processors (*see, e.g.*, FIG. 3A, item 51 and column 8, lines 58-65). Thus, Christensen fails to teach that each service mask is configured to indicate the next bit position in the current group to be examined for a set bit when the corresponding service class is the current service class.

Claim 28:

Regarding claim 28, Christensen fails to anticipate a system wherein the one service unit is a quantum smaller than a maximum message size for the channel requests.

The Examiner argues that Christensen's message size of the interrupt request SIGI exceeds the size of an interrupt request entry in the queues Q0-Q7, citing column 5, line 63 – column 6, line 50. However, this can only be speculation on the Examiner's part since Christensen is completely silent regarding the relative (or absolute) sizes of his interrupt request message SIGI and the size of an entry in his queues. The Examiner's cited passage only provides a general overview of Christensen's system but does not describe how one service unit is a quantum smaller than a maximum message size for channel requests. Nowhere does Christensen discuss or describe the sizes of either the interrupt request SIGI message or of the interrupt request entries in the queues Q0-Q7. Nor does Christensen define any sort of quantum or maximum message size for channel requests. There is no way, other than through hindsight-based speculation, to interpret the size relationship between the various requests and messages of Christensen's system. The Examiner is clearly using hindsight analysis and speculation in his interpretation of

the sizes of Christensen's interrupt requests, SIGI, and interrupt request entries in queues Q0-Q7. Furthermore, it is unclear how the Examiner's assertions are even relevant to the limitation of claim 28 when read in conjunction with claim 19 from which it depends.

Second Ground of Rejection:

Claims 12-17, 24 and 29-38 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Christensen. Appellants traverse this rejection for the following reasons. Different groups of claims are addressed under their respective subheadings.

In regard to independent claims 12, 29, 30 and 31, the Examiner cites *St. Regis Paper Co. v. Bemis Co., Inc.* for the premise that it is generally considered to be within the ordinary skill of the art to duplicate parts for a multiplied effect. However, duplicating queues Q0-Q7 in Christensen would simply provide more queues at the same level as queues Q0-Q7. Duplicating queues Q0-Q7 in Christensen would not create intermediate levels of the hierarchical channel map or an organization of channel sections. Furthermore, the one or more intermediate levels recited in claims 12, 29 and 30 are not "duplicate parts for a multiplied effect" of either the top level or lowest level of the hierarchical channel map. The one or more intermediate levels are recited as having functionality distinct from either the top-level or lowest level. Therefore, the Examiner's reliance on *St. Regis Paper Co. v. Bemis Co., Inc.* is clearly misplaced.

Appellant also respectfully disagrees with the Examiner's assertion that "the stages of Christensen provide for an organized system, and accordingly, at the time of the invention it would have been obvious to one of ordinary skill in the art to include additional organizational stages within the hierarchy of Christensen in order to provide additional organization for a multiplied effect." Christensen teaches eight queues, each of which may contain a number of active entries representing pending interrupt requests (col. 6, lines 3 – 14). Christensen further teaches that a pending register has eight bit positions that correspond to the eight queues. The set state of a pending register bit indicates that its queue is non-empty, while a reset state indicates its corresponding bit is

empty (Col. 6, lines 24 – 50). Christensen does not teach "organizational stages," as the Examiner suggests. To the extent that the PND register or queues in Christensen could be considered "organizational stages", any duplication of these parts would only have an effect at the same level. Accordingly, it would not have been obvious to introduce intermediate stages between Christensen's queues and pending register.

Additional distinctions for certain claims are discussed below.

Claim 12:

In regard to independent claim 12, Christensen does not teach selecting a set bit in a top level of a hierarchical channel map, wherein the set bit in the top level indicates a group of bits in a next level of the hierarchical channel map to examine. In Christensen, a channel controller 24 sends an SIGI command to system controller 22, "and the decoded command sets a corresponding bit in PND register 46 to identify the queue on which an entry is to be added" (Christensen -- col. 6, lines 31-34). Thus, the SIGI command from the channel controller 24 only identifies the queue on which an entry is to be added, not a group of bits in a next level of the hierarchical channel map to examine. Christensen teaches adding an entry to the queue, not examining a group of bits in a next level of the hierarchical channel map.

Christensen also teaches that a central processor 20, 21 selects from among active states of PND register 46 as broadcast by system controller 22 in order to accepts a queue to handle. However, the PND register position selected by the central processor only indicates a particular queue that the processor has accepted for handling. If the processor is selected by system controller 22, the processor then removes the oldest active entry on the accepted queue (col. 7, lines 20-24). Again, the selection of a PND register position by a processor in Christensen does not indicate a group of bits in a next level of a hierarchical channel map to examine. To meet the language of claim 12, the Examiner's position must be that a set bit in Christensen's PND register indicates a group of bits in a next level of the hierarchical channel map to examine. However, nothing in Christensen

examines bits in one of queues Q0-Q7 as a next level of a hierarchical channel map. Christensen only teaches that a set bit of the PND register indicates a queue for which an entry is either added or removed. However, adding or removing an entry from a queue has nothing to do with examining a group of bits in a next level of a hierarchical channel map.

Christensen also does not teach examining in each of one or more intermediate levels of the hierarchical channel map only a group of bits indicated by the set bit selected in the previous level and selecting a set bit from each examined group, wherein the top level is the previous level for a first intermediate level. The Examiner cites *St. Regis Paper Co. v. Bemis Co., Inc.* for the premise that it is generally considered to be within the ordinary skill of the art to duplicate parts for a multiplied effect. However, duplicating queues Q0-Q7 in Christensen would simply provide more queues at the same level as queues Q0-Q7. Duplicating queues Q0-Q7 in Christensen would not create intermediate levels of the hierarchical channel map or an organization of channel sections. Please refer to the discussion above for a more detail refutation of the Examiner's reliance on *St. Regis Paper Co. v. Bemis Co., Inc.* and regarding the Examiner's assertion regarding "organization stages" in Christensen's system.

In response to the above arguments, the Examiner states in his Response to Arguments section (and also in the Advisory Action) that "it would have been obvious to one of ordinary skill in the art to utilize the organization of the stages for a multiplied effect, and not to provide additional queues QO-Q7 in the same lowest level or a copy of one of the same levels as appellant has contested." However, **when one considers only the prior art**, the Examiner's statement clearly amounts to hindsight reasoning. The organization of a PND register having eight bits each corresponding to one of eight queues for holding pending interrupts does not in any way suggest "examining in each of one or more intermediate levels of the hierarchical channel map only a group of bits indicated by the set bit selected in the previous level and selecting a set bit from each examined group, wherein the top level is the previous level for a first intermediate level"

as recited in claim 12. Christensen's PND register and queues do not provide any suggestion of an intermediate level of a hierarchical channel map in which only a group of bits indicated by the set bit selected in the previous level are examined and from which a set bit is selected.

The Examiner also refers to the brief mention at the end of Christensen's disclosure that the concepts could be extended to a system having more than two central processors and more than one system controller. At most this would suggest having additional PND registers (one for each system controller). The presence of more central processors and system controllers in Christensen's system would not suggest an intermediate level of a hierarchical channel map in which only a group of bits indicated by the set bit selected in the previous level are examined and from which a set bit is selected. The Examiner speculates that an embodiment of Christensen's system having multiple system controllers (each having a PND register) would suggest an additional register having bit positions each corresponding to one of the system controllers. This is pure hindsight conjecture by the Examiner. Having multiple systems controllers does not suggest any such additional register level. At most, it would simply suggest having another PND register which would be at the same level as the current PND register.

Furthermore, Christensen does not teach examining in a lowest level of the hierarchical channel map only a group of bits indicated by the previous intermediate level and selecting a set bit from the examined group of the lowest level, wherein the selected bit at the lowest level indicates one of the plurality of communication channels to be serviced, and servicing a communication request from the communication channel indicated by the selected bit from the lowest level of the hierarchical channel map. As discussed above, Christensen only teaches that a set bit of the PND register indicates a queue for which an entry is either added or removed. To meet the language of claim 12, the Examiner must be asserting that something in Christensen examines a group of bits of one of the queues Q0-Q7 as indicated by the PND register and selects a set bit from the examined group of bits of the queues wherein the selected bit indicates one of a plurality

of communication channels to be serviced. However, Christensen only teaches adding and removing entries from queues Q0-Q7. Adding or removing an entry from a queue does not teach examining a group of bits in a lowest level of a hierarchical channel map and selecting a set bit from the examined group of bits wherein the selected bit indicates one of a plurality of communication channels to be serviced.

Claims 13:

Christensen additionally fails to teach or suggest selecting the next set bit from the group of bits in the lowest level from which the previous set bit was selected. Instead, Christensen teaches that entries in each queue Q0-Q7 are serviced in a first-in, first-out (FIFO) manner (column 3, lines 22-25) and that individual CPs select, or accept, the highest-priority, non-empty queue to service (column 6, lines 60-67). Christensen completely fails to mention anything regarding selecting the next set bit from the group of bits in the lowest level from which the previous set bit was selected.

Following the Examiner's interpretation of Christensen, which (erroneously) equates each of queues Q0-Q7 as a separate grouping of bits in the lowest level of a hierarchical channel map, Christensen would have to select a next entry from the same queue from which the previous entry was serviced. However, as described above, Christensen does not teach the consecutive servicing of multiple entries from the same queue. Instead, Christensen's system broadcasts, to each of multiple CPs, a set of bits each corresponding to one of the queues. A CP then accepts and services exactly one entry from one of the queues. Subsequent entries are services in the same manner – by broadcasting bits identifying non-empty queues and a CP accepting and servicing one of the queues. Christensen explains this behavior very simply at column 4, lines 16-25 by describing how after a CP completes it's processing of one entry in the accepted queue, "all processors may contend for the next entry on the queue."

Additionally, Christensen fails to teach or suggest if no more bits are set in that group, selecting the next set bit from the next higher level in the same group as the

previous selected bit at that level or repeating said selecting the next set bit from the next higher level until a set bit is found and then selecting a set bit from each group at each lower level as indicated by the set bit at the previous level. Instead, Christensen teaches that after removing and servicing an entry from one of queues Q0-Q7 if a CP determines that the queue is empty, "the CP sends a reset signal on a special line to the CI controller" and that the reset signal "causes the CI controller to reset the corresponding queue position in the pending register in order to broadcast that there are no other pending interrupt entries in that queue at that time" (column 4, lines 30-47).

The Examiner does not point out any portion of Christensen that describes, nor does Christensen mention anywhere, selecting the next set bit from the next higher level (of the hierarchical channel map) in the same group as the previous selected bit at that level. Instead, Christensen teaches only that once a CP has services that last entry from a queue the corresponding bit in the PDN register is set to zero. As described above, Christensen does not teach examining the hierarchical channel map to determine a next set bit corresponding to a channel that should be serviced next. Instead, Christensen's CI controller selects which central processor should service the oldest (first-in/first-out) entry from the highest priority queue.

Christensen also fails to teach servicing subsequent communication requests during each subsequent communication time from the communication channel indicated by the selected bit from the lowest level of the hierarchical channel map for each subsequent service time. Instead, as described herein above, Christensen teaches that the CI controller selects which central processor should service the oldest (first-in/first-out) entry from the highest priority queue. Christensen does not teach that the CI controller selects which queue should be serviced based on a selected bit from within queues Q0-Q7 (interpreted as the lowest level of a hierarchical channel map by the Examiner).

Claim 14:

In regard to claim 14, Christensen does not teach setting a bit in the lowest level of the hierarchical channel map for each communication channel that has a pending communication request. The Examiner maintains that Christensen's interrupt request queues Q0-Q7 are a lowest level in a hierarchical channel map and also contends that storing an interrupt request entry in one of queues Q0-Q7 is setting a bit in the lowest level of a hierarchical channel map. However, this is not a correct interpretation of Christensen's queues. Adding an interrupt request entry is very different from setting a single bit in a channel map. Christensen does teach that an interrupt request entry is a single bit. Please see the arguments above regarding claims 12 and 18 for a discussion of why the Examiner's interpretation of Christensen's queues Q0-Q7 is incorrect.

Claim 15:

The Examiner, in his rejection of claim 15, has referred to limitations different from those recited in claim 15. Therefore, the rejection of claim 15 is improper.

Additionally, Christensen does not teach updating a service mask for each level of the hierarchical channel map to indicate the position of the last selected set bit within the indicated group of bits. The Examiner contends that Christensen's I/O mask 51 is a service mask. However, I/O mask 51, as argued above, does not indicate the position of any bits anywhere. Instead, I/O mask 51 is used to prevent a particular central processor from accepting and servicing particular ones of queues Q0-Q7. Additionally, as argued above, Christensen teaches that the acceptance circuit for each central processor includes an I/O mask 51 and does not teach including an I/O mask 51 for each level of a hierarchical channel map.

Christensen also fails to teach wherein each time an indicated group of bits is examined to select the next set bit, it is examined starting at the next bit position from the position indicated by the corresponding service mask. As discussed above, Christensen's I/O mask 51 does not indicate any bit position in any group of bits. Instead, I/O mask 51 is used to limit which of queues Q0-Q7 may be serviced by a particular central processor.

Please refer to the discussions above regarding claims 20, 21, 25, and 27 for arguments rebutting the Examiner's interpretation of Christensen's I/O mask 51 as a service mask.

Claim 16:

The Examiner, in his rejection of claim 16, has again referred to limitations different from those recited in the rejected claim. Thus, the rejection of claim 16 is improper.

Furthermore, Christensen does not teach wherein each bit position of the top level of the hierarchical channel map represents a different section of the plurality of communication channels. The Examiner erroneously contends that Christensen's PND register is a top level of a hierarchical channel map. Please refer the discussion regarding claims 12 and 18 above for arguments rebutting the Examiner's interpretation of the PND register as a top level of a hierarchical channel map. Also, Christensen's queues do not represent different sections of a plurality of communication channels.

Christensen additionally fails to teach mapping each section of communication channels to a different service class. Nowhere does Christensen mention service classes or mapping each section of communication channels to a different service class. The Examiner argues that I/O mask 51 represents a masking of different service classes. However, as argued above, I/O mask 51 has nothing to do with services classes or with mapping sections of communication channels. Instead, I/O mask 51 is used to limit which of queues Q0-Q7 may be serviced by a particular central processor. Please refer to the discussions above regarding claims 20, 21, 25, and 27 for arguments rebutting the Examiner's interpretation of Christensen's I/O mask 51 as a service mask.

Christensen additionally fails to teach indicating in a service array an order in which each service class is to be serviced; and choosing a service class according to said

service array. The Examiner argues that I/O mask 51 is also a service array indicating an order in which to service each service class. However, as noted above, the Examiner argues both that I/O mask 51 represents different service masks and that it represents different service *class* masks. Not only does each of the Examiner's interpretations of I/O mask 51 conflict with Christensen's teaching regarding I/O mask 51 (see above), but additionally, the Examiner is arguing two inconsistent interpretations of a single component of Christensen's system.

Furthermore, Christensen also does not teach examining only portions of the hierarchical channel map corresponding to sections of communication channels mapped to the chosen service class to select the next set bit at the lowest level. Please refer to the arguments presented above regarding claims 25 and 26, which apply to claim 16 as well.

Claim 17:

The Examiner, in this rejection of claim 17, has not presented arguments in reference the limitations recited in claim 17, but instead has presented argument in reference to the limitations recited in a different claim. Therefore, the rejection of claim 17 is improper.

Furthermore, Christensen fails to teach updating a service mask for each level of the hierarchical channel map and for each service class to indicate the position of the last selected set bit within the indicated group of bits and for the corresponding service class, wherein each time an indicated group of bits is examined to select the next set bit, it is examined starting at the next bit position from the position indicated by the corresponding service mask for the current service class. Please refer to the arguments presented above regarding claims 25, 26 and 27, which apply to claim 17 as well.

Claim 29:

In regard to claim 29, Christensen does not teach a top-level service mask, a second level service mask and a bottom level service mask and the corresponding operations as recited in claim 29. The Examiner only refers to Christensen's I/O mask 51 and col. 6, lines 51-67. Christensen's I/O mask 51 does not have any relevance to the top level, second level and bottom level service masks recited in claim 29. Christensen's I/O mask 51 is described only as preventing the processors from accepting a particular queue for handling after the particular queue has already been accepted by another processor. Furthermore, Christensen's I/O mask 51 does not indicate a position in a group of bits in a level of a hierarchical channel map. Nor is any bit in I/O mask 51 set to indicate a selected first set bit in the current level group. Instead, bits in I/O mask 51 are set to indicate that a particular processor should not service entries from a one or more of queues Q0-Q7. (See, column 6, lines 56-67). Furthermore, no bit position is every indicated by I/O mask 51.

Moreover, claim 29 recites **three different service masks**. The single I/O mask in Christensen clearly does not teach three different service masks as recited in claim 29. Instead, Christensen teaches a separate I/O mask 51 in the interrupt acceptance circuits for each of multiple central processors (FIG 3A, item 51 and column 8, lines 58-65).

Furthermore, claim 29 recites that each service mask is set to indicate the position of the selected bit in the accessed corresponding level group. **The I/O mask in Christensen has nothing to do with indicating a position of a selected bit of a corresponding level bit group of a hierarchical channel map.**

Nor does Christensen teach top level, second level and bottom level groups of a hierarchical channel map as recited in claim 29. As discussed above, Christensen only teaches a PND register having eight bits each corresponding to on of queues Q0-Q7.

Claim 30:

In regard to independent claim 30, Christensen does not teach, for a first service time, examining a portion of one or more intermediate levels of the hierarchical channel map to select a lowest level group of the communication channels in which at least one channel has a pending communication request, wherein each intermediate level indicates for each of a plurality of groups of the communication channels if at least one channel of that group has a pending communication request, wherein the groups at each intermediate level are sized smaller than at the previous intermediate level, and wherein examining a portion of each intermediate level determines which portion of the next hierarchical channel map level to examine.

As discussed above, Christensen only teaches a PND register having eight bits each corresponding to one of queues Q0-Q7 for which entries are added or removed. As discussed above, Christensen does not teach, or even mention, a hierarchical channel map. The Examiner argues that Christensen's PDN register, in conjunction with queues Q0-Q7, make up a hierarchical channel map. Please refer to the discussions above regarding claims 18 and 12 for a rebuttal of the Examiner's argument. Furthermore, nowhere does Christensen discuss any intermediate level between the PND register and queues Q0-Q7. Hence, even following the Examiner's line of reasoning, Christensen fails to teach examining a portion of one or more intermediate levels of the hierarchical channel map.

The Examiner cites *St. Regis Paper Co. v. Bemis Co., Inc.* for the premise that it is generally considered to be within the ordinary skill of the art to duplicate parts for a multiplied effect. However, duplicating queues Q0-Q7 in Christensen would simply provide more queues at the same level as queues Q0-Q7. Duplicating queues Q0-Q7 in Christensen would not create intermediate levels of the hierarchical channel map or an organization of channel sections. Furthermore, the one or more intermediate levels recited in claims 12, 29 and 30 are not "duplicate parts for a multiplied effect" of either the top level or lowest level of the hierarchical channel map. The one or more intermediate levels are recited as having functionality distinct from either the top-level or

lowest level. Please refer to the discussion regarding claims 12, 29, 30, and 31 at the beginning of the Second Ground of Rejection for a detailed argument rebutting the Examiner's reliance on *St. Regis Paper Co. v. Bemis Co., Inc.*

Claims 31, 37 and 38:

In regard to independent claim 31, Christensen does not teach a second memory configured to store a top level of a hierarchical channel map, wherein the plurality of communication channels are organized in channel sections, and wherein for each channel section said top level indicates if at least one of the communication channels within that section has a pending communication request. As argued above, the PND register in Christensen only indicates which queues have pending interrupt requests. However, this does not serve to organize the communication channels in Christensen into channel sections as recited in claim 31.

Further in regard to claim 31, Christensen does not teach a host adapter configured to determine a next channel to service by examining the lowest level in the first memory, wherein the host adapter determines the next channel to service by examining no more of the lowest level than a portion of the lowest level corresponding to one of the channel sections indicated by the top level as having at least one pending communication request. As discussed above, nothing in Christensen examines queues Q0-Q7 to determine a next channel to service. Instead, the entries in a queue are removed and serviced in a first-in, first-out order. Thus, there is no reason to examine Christensen's queues to determine a next channel to service.

Please refer to the discussions above regarding claims 12 and 18 for more detailed arguments that also apply to claim 31.

Claim 32:

Regarding claim 32, Christensen does not teach a system wherein the one or more

levels stored in said first memory further comprises an intermediate level of said hierarchical channel map, wherein the communication channels are further organized into channel groups, wherein for each channel group the intermediate level indicates if at least one of the communication channels within that group has a pending communication request, and wherein said host adapter is further configured to determine the next channel to service by examining no more of the lowest level than a portion of the lowest level corresponding to one of the channel groups indicated by the intermediate level as having at least one pending communication request, wherein each channel section includes a plurality of channel groups.

As discussed above, Christensen does not teach a hierarchical channel map, nor does he teach or suggest any intermediate levels in a hierarchical channel map. Furthermore, as also discussed above, Christensen does not teach examining any lowest level of a hierarchical channel map. The Examiner argues that Christensen's queues Q0-Q7 are a lowest level of a hierarchical channel map, however, even following the Examiner's line of reasoning (please refer to the discussions above regarding claims 12 and 18 for detailed arguments regarding the Examiner's interpretation of queues Q0-Q7), Christensen fails to teach or suggest examining no more of lowest level than a portion of the lowest level corresponding to one of the channel groups indicated by the intermediate level. There is no need to examine queues Q0-Q7 because Christensen teaches that the entries of queues Q0-Q7 are serviced in a first-in first-out manner. The Examiner has failed to present any argument providing a motivation to modify Christensen to include examining queues Q0-Q7 to determine a next channel to service rather than servicing the entries in a first-in first-out manner as taught by Christensen.

Claim 33:

Regarding claim 33, Christensen does not teach a system wherein the top level stored in the second memory is further configured to indicate which portions of the intermediate level contain at least one indication that at least one channel within one of the channel groups has a pending communication request.

The Examiner does not provide any argument nor does he cite any portion of Christensen for his rejection of claim 33. The Examiner argues only that it is generally considered to be within the ordinary skill of the art to duplicate parts for a multiplied effect citing *St. Regis Paper Co. v. Bemis Co., Inc.* However, as argued above, even if one could duplicate the PDN register and/or the queues in Christensen, the result would still not teach wherein the top level is configured to indicate which portions of the intermediate level contain at least one indication that at least one channel within one of the channel groups has a pending communication request. Christensen's PND register only indicates which queues have interrupt request entries needing to be serviced. Duplicating the PDN register would not change the workings of the PDN register to indicate which portions of an intermediate level of a hierarchical channel map, as recited in claim 33. Instead, duplicating the PDN register would result in two registers indicating which of queues Q0-Q7 are not empty.

Please refer to the discussion regarding claims 12, 29, 30, and 31 at the beginning of the Second Ground of Rejection for a detailed argument rebutting the Examiner's reliance on *St. Regis Paper Co. v. Bemis Co., Inc.*

Claim 34:

Regarding claim 34, Christensen does not teach a system wherein the lowest level comprises a plurality of low level bits, wherein each low level bit is set if a corresponding one of said plurality of channels has a pending communication request and is cleared if the corresponding one of said plurality of channels does not have a pending communication request. The Examiner has failed to provide any specific arguments regarding his rejection of claim 34, except to assert that it is generally considered to be within the ordinary skill of the art to duplicate parts for a multiplied effect citing *St. Regis Paper Co. v. Bemis Co., Inc.* Please refer to the discussion regarding claims 12, 29, 30, and 31 at the beginning of the Second Ground of Rejection for a detailed argument rebutting the Examiner's reliance on *St. Regis Paper Co. v. Bemis Co., Inc.*

Furthermore, the Examiner contends that Christensen's queues Q0-Q7 are the lowest level of a hierarchical channel map. However, Christensen teaches that interrupt request entries are stored and removed from queues Q0-Q7. Christensen does not mention that each bit of queues Q0-Q7 is set if the channel to which it is mapped has a pending communication request and is cleared if not. There is no setting and clearing of bits in Christensen's queues. Under Christensen's system, there is no single specific bit that reflects the state (e.g. whether there is a pending interrupt request) of any particular channel. Instead, Christensen's queues function purely and only as queues. Entries are added as interrupts occur and removed as they are serviced.

Claim 35:

Regarding claim 35, Christensen does not teach a system wherein the one or more levels stored in the first memory further comprises an intermediate level comprising a plurality of intermediate level bits, wherein each intermediate level bit corresponds to a group of low level bits, and wherein each intermediate level bit is set if its corresponding group of low level bits has at least one bit set and is cleared if no bits are set in its corresponding group of low level bits.

The Examiner has failed to provide any specific arguments regarding his rejection of claim 35, except to assert that it is generally considered to be within the ordinary skill of the art to duplicate parts for a multiplied effect citing *St. Regis Paper Co. v. Bemis Co., Inc.* Please refer to the discussion regarding claims 12, 29, 30, and 31 at the beginning of the Second Ground of Rejection for a detailed argument rebutting the Examiner's reliance on *St. Regis Paper Co. v. Bemis Co., Inc.*

Additionally, any duplication of either Christensen's queues or the PND register would fail to result in an intermediate level comprising a plurality of intermediate level bits, wherein each intermediate level bit corresponds to a group of low level bits. Instead, duplicating either Christensen's queues or the PND register, or both, would result in a

system that had additional queues to store interrupt request entries and/or additional registers to indicate whether or not those queues are empty. Duplicating these parts of Christensen's system would not change the manner in which Christensen stores interrupt request entries (e.g. in queues Q0-Q7), nor would it change the manner in which Christensen indicates whether there are any interrupt request entries needing services (e.g. with the PDN register).

Claim 36:

Regarding claim 36, Christensen does not teach a system wherein the top level comprises a plurality of top level bits, wherein each top level bit corresponds to a group of intermediate level bits, and wherein each top level bit is set if its corresponding group of intermediate level bits has at least one bit set and is cleared if no bits are set in its corresponding group of intermediate level bits.

As with claims 30-35, discussed above, the Examiner has failed to provide any specific arguments regarding his rejection of claim 36, except to assert that it is generally considered to be within the ordinary skill of the art to duplicate parts for a multiplied effect citing *St. Regis Paper Co. v. Bemis Co., Inc.* Please refer to the discussion regarding claims 12, 29, 30, and 31 at the beginning of the Second Ground of Rejection for a detailed argument rebutting the Examiner's reliance on *St. Regis Paper Co. v. Bemis Co., Inc.*

Furthermore, any duplication of either Christensen's queues or the PND register would fail to result each top level bit corresponding to a group of intermediate level bits. Instead, duplicating either Christensen's queues or the PND register, or both, would result in a system that had additional queues to store interrupt request entries and/or additional registers to indicate whether or not those queues are empty. Duplicating these parts of Christensen's system would not change the manner in which Christensen stores interrupt request entries (e.g. in queues Q0-Q7), nor would it change the manner in which

IX. CLAIMS APPENDIX

The claims on appeal are as follows.

12. A method for servicing a plurality of communication channels, comprising:

for a first service time:

selecting a set bit in a top level of a hierarchical channel map, wherein the set bit in the top level indicates a group of bits in a next level of the hierarchical channel map to examine;

examining in each of one or more intermediate levels of the hierarchical channel map only a group of bits indicated by the set bit selected in the previous level and selecting a set bit from each examined group, wherein said top level is the previous level for a first intermediate level;

examining in a lowest level of the hierarchical channel map only a group of bits indicated by the previous intermediate level and selecting a set bit from the examined group of the lowest level, wherein the selected bit at the lowest level indicates one of the plurality of communication channels to be serviced; and

servicing a communication request from the communication channel indicated by the selected bit from the lowest level of the hierarchical channel map.

13. The method as recited in claim 12, further comprising:

for subsequent service times:

selecting the next set bit from the group of bits in the lowest level from which the previous set bit was selected, or if no more bits are set in that group, selecting the next set bit from the next higher level in the same group as the previous selected bit at that level or repeating said selecting the next set bit from the next higher level until a set bit is found and then selecting a set bit from each group at each lower level as indicated by the set bit at the previous level; and

servicing subsequent communication requests during each subsequent communication time from the communication channel indicated by the selected bit from the lowest level of the hierarchical channel map for each subsequent service time.

14. The method as recited in claim 12, further comprising:

setting a bit in the lowest level of the hierarchical channel map for each communication channel that has a pending communication request; and

setting a bit at each higher level of the hierarchical channel map to indicate that a corresponding group of bits in the next lower level has at least one set bit.

15. The method as recited in claim 12, further comprising updating a service mask for each level of the hierarchical channel map to indicate the position of the last selected set bit within the indicated group of bits, wherein each time an indicated group of bits is examined to select the next set bit, it is examined starting at the next bit position from the position indicated by the corresponding service mask.

16. The method as recited in claim 13, wherein each bit position of the top level of the hierarchical channel map represents a different section of the plurality of communication channels, the method further comprising:

mapping each section of communication channels to a different service class;

indicating in a service array an order in which each service class is to be serviced;

choosing a service class according to said service array; and

examining only portions of the hierarchical channel map corresponding to sections of communication channels mapped to the chosen service class to select the next set bit at the lowest level.

17. The method as recited in claim 16, further comprising updating a service mask for each level of the hierarchical channel map and for each service class to indicate the position of the last selected set bit within the indicated group of bits and for the corresponding service class, wherein each time an indicated group of bits is examined to select the next set bit, it is examined starting at the next bit position from the position indicated by the corresponding service mask for the current service class.

18. A system for servicing communication queues, comprising:

memory configured to store a hierarchical channel map comprising a plurality of levels, wherein each bit of the lowest level is mapped only to a different one of a plurality of communication channels, and wherein each bit of each higher level is mapped to a group of bits at the next lower level;

a host adapter configured to maintain the hierarchical channel map, wherein each bit at the lowest level is set if the channel to which it is mapped has a

pending communication request and is cleared if not, and wherein each bit of each higher level is set if at least one bit is set in the lower level group to which it is mapped and is cleared if not; and

wherein said host adapter is configured to examine the hierarchical channel map to determine a next one of the communication channels to service.

19. The system as recited in claim 18, wherein said host adapter is configured to service for one service unit a channel request from a channel mapped to a set bit at the lowest level of the hierarchical channel map, wherein the set bit is selected by examining a current group of bits at the lowest level of the hierarchical channel map to select a next set bit in that group indicating a channel with a pending request, and if no more bits are set in the current group, examining a current group at the next higher level to select a next set bit and then examining the next lower level group indicated by the selected higher level set bit.

20. The system as recited in claim 19, further comprising a service mask for each level of the hierarchical channel map, wherein each service mask is configured to indicate the next bit position in the current group to be examined for a set bit.

21. The system as recited in claim 20, wherein each service mask is configured to indicate the bit position within the current group for the corresponding level of the last selected set bit in that group, wherein the host adapter is configured to examine each group for the next set bit after the bit position indicated by the corresponding service mask.

22. The system as recited in claim 18, wherein each group of bits at one level of the hierarchical channel map has the same number of bits.

23. The system as recited in claim 18, wherein each group of bits at the lowest

level is accessible by a single memory access.

24. The system as recited in claim 18, wherein the top level of the hierarchical channel map is stored within a register comprised within the same integrated circuit as said host adapter.

25. The system as recited in claim 18, wherein said host adapter is configured to service channel requests according to service classes, wherein each channel is mapped to one or more service classes, and wherein the next channel to be serviced is selected by examining only portions of said hierarchical channel map corresponding to channels mapped to a current service class.

26. The system as recited in claim 25, wherein the current service class is selected according to an order of service classes indicated by a service array.

27. The system as recited in claim 25, further comprising one service mask per level of the hierarchical channel map per service class, wherein each service mask is configured to indicate the next bit position in the current group to be examined for a set bit when the corresponding service class is the current service class.

28. The system as recited in claim 19, wherein said one service unit is smaller than a maximum message size for the channel requests.

29. A computer readable medium comprising program instructions, wherein said program instructions are operable to:

select a first bit set in a current top level group of a hierarchical channel map after a position indicated by a top level service mask;

set the top level service mask to indicate the position of the selected first set bit in

the current top level group;

in a second level of the hierarchical channel map, access a second level group indicated by the selected set bit from the top level;

select a first bit set in the accessed second level group after a position indicated by a second level service mask;

set the second level service mask to indicate the position of the selected bit in the accessed second level group;

in a bottom level of the hierarchical channel map, access a bottom level group indicated by the selected set bit from the second level;

select a first bit set in the accessed bottom level group after a position indicated by a bottom level service mask;

set the bottom level service mask to indicate the position of the selected bit in the accessed bottom level group; and

service a request from a channel indicated by the selected set bit from the bottom level.

30. A method for servicing a plurality of communication channels, comprising:

for a first service time, examining a top level of a hierarchical channel map to select a section of the communication channels in which at least one channel has a pending communication request, wherein said top level indicates for each of a plurality of sections of the communication channels

if at least one channel of that section has a pending communication request;

for said first service time, examining a portion of one or more intermediate levels of the hierarchical channel map to select a lowest level group of the communication channels in which at least one channel has a pending communication request, wherein each intermediate level indicates for each of a plurality of groups of the communication channels if at least one channel of that group has a pending communication request, wherein the groups at each intermediate level are sized smaller than at the previous intermediate level, and wherein examining a portion of each intermediate level determines which portion of the next hierarchical channel map level to examine; and

selecting a next communication channel to be serviced from the lowest level group selected by said examining a portion of one or more intermediate levels.

31. A system for servicing a plurality of communication channels, comprising:

a first memory configured to store one or more levels of a hierarchical channel map, wherein said one or more levels comprises a lowest level for storing an indication of which ones of the plurality of communication channels have pending communication requests;

a second memory configured to store a top level of said hierarchical channel map, wherein the plurality of communication channels are organized in channel sections, and wherein for each channel section said top level indicates if at least one of the communication channels within that section has a pending communication request; and

a host adapter configured to determine a next channel to service by examining said lowest level in said first memory, wherein said host adapter determines the next channel to service by examining no more of said lowest level than a portion of said lowest level corresponding to one of said channel sections indicated by said top level as having at least one pending communication request.

32. The system as recited in claim 31, wherein said one or more levels stored in said first memory further comprises an intermediate level of said hierarchical channel map, wherein said communication channels are further organized into channel groups, wherein for each channel group said intermediate level indicates if at least one of the communication channels within that group has a pending communication request, and wherein said host adapter is further configured to determine the next channel to service by examining no more of said lowest level than a portion of said lowest level corresponding to one of said channel groups indicated by said intermediate level as having at least one pending communication request, wherein each channel section includes a plurality of channel groups.

33. The system as recited in claim 32, wherein said top level stored in said second memory is further configured to indicate which portions of said intermediate level contain at least one indication that at least one channel within one of said channel groups has a pending communication request.

34. The system as recited in claim 31, wherein said lowest level comprises a plurality of low level bits, wherein each low level bit is set if a corresponding one of said plurality of channels has a pending communication request and is cleared if the corresponding one of said plurality of channels does not have a pending communication request.

35. The system as recited in claim 34, wherein said one or more levels stored in said first memory further comprises an intermediate level comprising a plurality of intermediate level bits, wherein each intermediate level bit corresponds to a group of low level bits, and wherein each intermediate level bit is set if its corresponding group of low level bits has at least one bit set and is cleared if no bits are set in its corresponding group of low level bits.

36. The system as recited in claim 35, wherein said top level comprises a plurality of top level bits, wherein each top level bit corresponds to a group of intermediate level bits, and wherein each top level bit is set if its corresponding group of intermediate level bits has at least one bit set and is cleared if no bits are set in its corresponding group of intermediate level bits.

37. The system as recited in claim 31, wherein said first memory and said second memory are part of a memory block accessible by said host adapter.

38. The system as recited in claim 31, wherein said second memory is a register comprised within an integrated circuit with said host adapter.

X. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.